Interfacing Servo Motor with ARM (Firebird V)

Joel M. Pinto and Vishal H. Rajai
eYantra Summer Internship'14
Embedded Real-Time Systems Lab
Indian Institute of Technology, Bombay

IIT Bombay
July 21, 2014

Agenda of Discussion

1. Introduction
2. Servo motor
   - Principle and Working
   - Operating Servo Motor
   - Selection of Servo motor
3. ARM7 LPC2148
   - Generating PWM signals
   - Code

Prerequisite knowledge

1. Basic IO Interfacing using ports
2. Working with PWM registers of ARM7 LPC2148.

Introduction

- Servo motors (or servos) are self-contained devices that rotate or push parts of a machine with great precision.
- Servos can put out about 42 oz/in of torque.
- Relatively inexpensive.
- Widely used for educational purpose in mechatronics as they can be controlled by a microcontroller.

Principle and Working

- A servo mainly contains:

Principle and Working

- A servo mainly contains:
  - dc motor

Principle and Working

- A servo mainly contains:
  - dc motor
  - gear train

Principle and Working

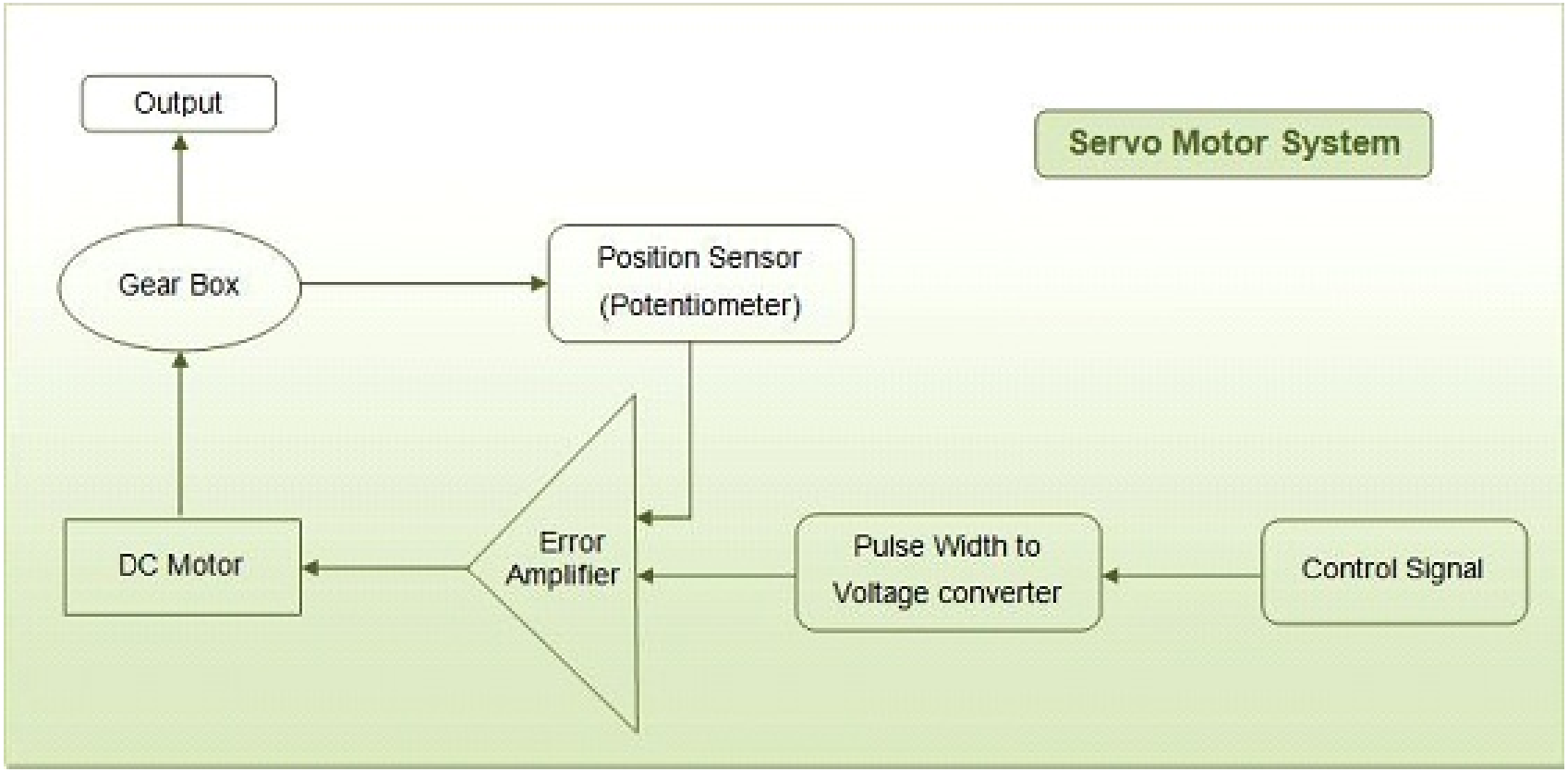
- A servo mainly contains:
  - dc motor
  - gear train
  - potentiometer

Principle and Working

- A servo mainly contains:
  - dc motor
  - gear train
  - potentiometer
  - control circuitry.

Principle and Working

- A servo mainly contains:
  - dc motor
  - gear train
  - potentiometer
  - control circuitry.
- Forming closed loop control system.


Servo Motor System
Output
Gear Box
Position Sensor
(Potentiometer)
DC Motor
Error
Amplifier
Pulse Width to
Voltage converter
Control Signal

- As shown in figure, control signal and output signal are fed to error detector.

- As shown in figure, control signal and output signal are fed to error detector.
- Resultant signal from error detector acts as input to the dc motor to rotate.

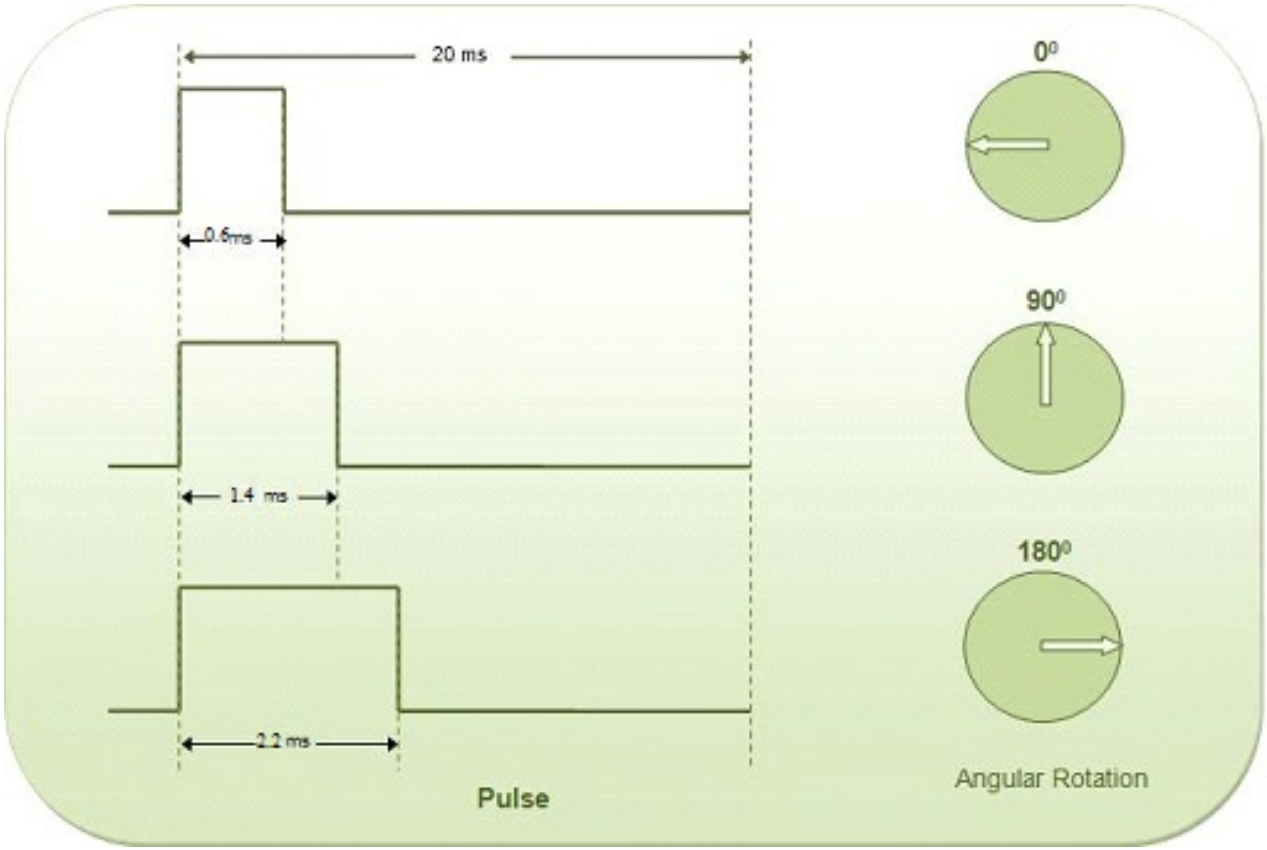
- As shown in figure, control signal and output signal are fed to error detector.
- Resultant signal from error detector acts as input to the dc motor to rotate.
- Also rotating the potentiometer knob coupled with its shaft via gears.

- As shown in figure, control signal and output signal are fed to error detector.
- Resultant signal from error detector acts as input to the dc motor to rotate.
- Also rotating the potentiometer knob coupled with its shaft via gears.
- Reaching desired angle, there would not be any difference in the signals fed to error detector.

- As shown in figure, control signal and output signal are fed to error detector.
- Resultant signal from error detector acts as input to the dc motor to rotate.
- Also rotating the potentiometer knob coupled with its shaft via gears.
- Reaching desired angle, there would not be any difference in the signals fed to error detector.
- Resulting in motor to stop rotating and wait at that position

Operating servo motor

- 'on-time' of a PWM signal is used as control signal to rotate motor at particular angle.



- This time period depends on the servo used and not on total time period or duty cycle of PWM signal.

- This time period depends on the servo used and not on total time period or duty cycle of PWM signal.
- Values of on-time for rotating servo to 0 and 180 degree are provided by the manufacturer in data sheet.

- This time period depends on the servo used and not on total time period or duty cycle of PWM signal.
- Values of on-time for rotating servo to 0 and 180 degree are provided by the manufacturer in data sheet.
- Graph of on-time period vs. angle is linear.

- This time period depends on the servo used and not on total time period or duty cycle of PWM signal.
- Values of on-time for rotating servo to 0 and 180 degree are provided by the manufacturer in data sheet.
- Graph of on-time period vs. angle is linear.
- Range of PWM frequency for operating servo is 40-60 Hz

Selection of Servo motor

Servo is selected based on its:

Selection of Servo motor

Servo is selected based on its:

- Torque

Selection of Servo motor

Servo is selected based on its:

- Torque
- Speed

Selection of Servo motor

Servo is selected based on its:

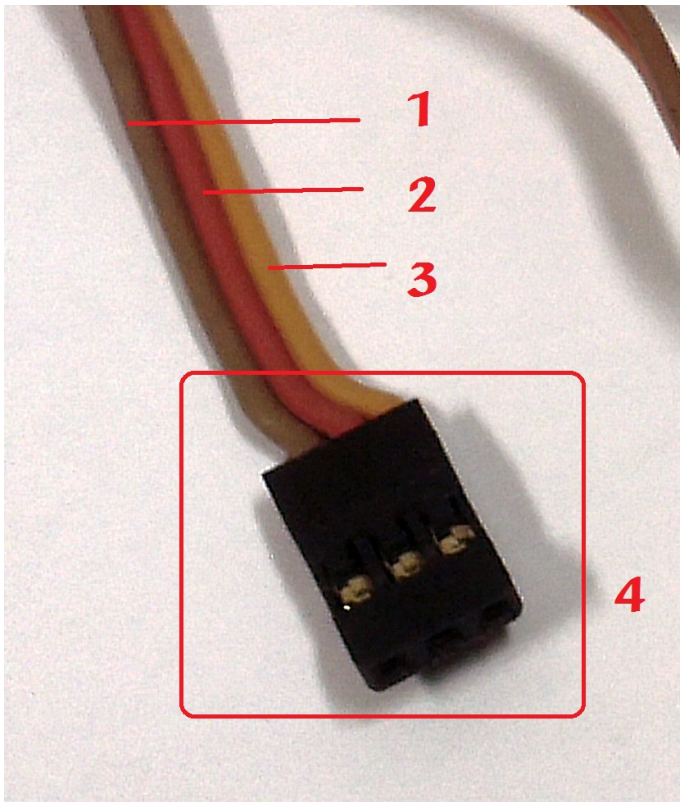
- Torque
- Speed
- Weight

Selection of Servo motor

Servo is selected based on its:

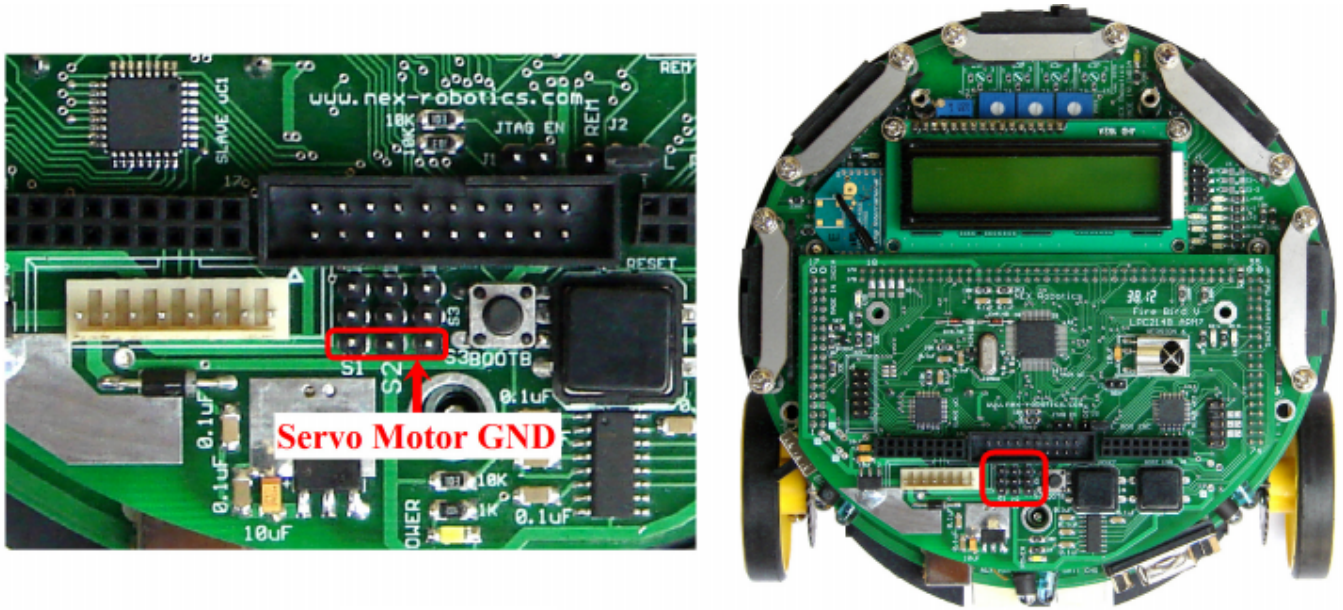
- Torque
- Speed
- Weight
- Dimensions

Servo connector

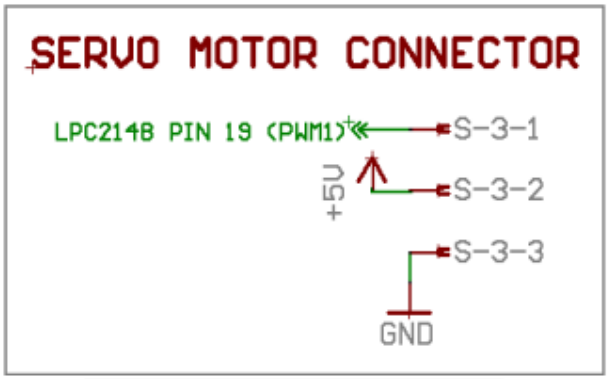


- 1– Ground
- 2– Power
- 3– Control signal
- 4– Connector to controller

Interfacing servo with Firebird V



Servo Connectors location



Servo Connectors Schematic

Using PWM register for PWM generation

Single edged PWM is used here to rotate servo motor.

- Let the Prescaler, PWMPR=120

Using PWM register for PWM generation

Single edged PWM is used here to rotate servo motor.

- Let the Prescaler, PWMPR=120
- $f_{cpu} = 12000000Hz$

Using PWM register for PWM generation

Single edged PWM is used here to rotate servo motor.

- Let the Prescaler, PWMPR=120
- $f_{cpu} = 12000000Hz$
- Let the frequency of the PWM signal be = 50 Hz

Using PWM register for PWM generation

Single edged PWM is used here to rotate servo motor.

- Let the Prescaler, PWMPR=120
- $f_{cpu} = 12000000Hz$
- Let the frequency of the PWM signal be = 50 Hz That gives, $PWMMR0 = 2000$

Using PWM register for PWM generation

Single edged PWM is used here to rotate servo motor.

- Let the Prescaler, PWMPR=120
- $f_{cpu} = 12000000Hz$
- Let the frequency of the PWM signal be = 50 Hz That gives, $PWMMR0 = 2000$
- Values of registers like PWMTCR and PINSEL can be found using ARM7 LPC2148 datasheet.

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120
  That increments counter every 120/12000000 = 0.01 ms

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120
  That increments counter every 120/12000000 = 0.01 ms
- Counts to be set to get 0.6 ms on-time period to rotate servo to 0°will be = 0.6/0.01 = 60

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120
  That increments counter every 120/12000000 = 0.01 ms
- Counts to be set to get 0.6 ms on-time period to rotate servo to 0°will be = 0.6/0.01 = 60 Similarly, for 180°, count value = 220

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120
  That increments counter every 120/12000000 = 0.01 ms
- Counts to be set to get 0.6 ms on-time period to rotate servo to 0°will be = 0.6/0.01 = 60 Similarly, for 180°, count value = 220
- As the relation between on-time period and corresponding degree is linear, so is the relation between the count value in PWMMR1 and degree

Using PWM register for PWM generation contd.

- Calculating value of PWMMR1:
  Prescaler = 120
  That increments counter every 120/12000000 = 0.01 ms
- Counts to be set to get 0.6 ms on-time period to rotate servo to 0°will be = 0.6/0.01 = 60 Similarly, for 180°, count value = 220
- As the relation between on-time period and corresponding degree is linear, so is the relation between the count value in PWMMR1 and degree
  That gives,
  $PWMMR1 = (degrees/1.125) + 60.0$

Code
Simple code

Code
Simple code

Header files

Code
Simple code

Header files

```
#include  <lpc214x.h>
```

Code
Simple code

Header files

```
#include  <lpc214x.h>
```

Main Logic

Code
Simple code

Header files

```
#include  <lpc214x.h>
```

Main Logic

```
float PositionPanServo = 0;
PositionPanServo = ((float)degrees / 1.125) + 60.0;
UpdateServoPos((unsigned int)PositionPanServo);
```

Thank You!

Post your queries on: helpdesk@e-yantra.org